

Amendments to the Claims

1. (Currently Amended) A general computer network controller for a
2 network node, coupled to a system area network, said controller comprising:
a network protocol engine configured to schedule packets for transmission onto
4 the system area network;
a data buffer configured to handle one or more payloads;
6 a fully associative context block configured to hold a plurality of last recently
used contexts to provide a dynamic resource allocation scheme reflecting run time
8 situations;
an address translation table coupled to said network protocol engine and
10 configured to:
maintain inbound address mapping; and
12 store context information not currently stored in said context block; and
a dedicated, programmable micro sequencer tightly coupled to said context block
14 and configured to:
control said context block; and
16 handle control flow and process multiple types of network packets and
protocols;
18 wherein said micro sequencer is packet format independent and network
independent; and
20 wherein said contexts are updated by said micro sequencer, by an inbound
scheduler and by said ~~a~~ network protocol engine.

2. (Previously Presented) The computer network controller of claim 1, further
2 comprising:

a scalable memory array configured as a table for Inbound address mapping of
4 registered memory and access protection, and further configured as a means for keeping
context information about all active channels.

3. (Previously Presented) The computer network controller of claim 1,

2 wherein said fully associative context block couples said inbound scheduler and said
network protocol engine, thereby facilitating an ability of said network controller to
4 pipeline tasks and execute in parallel.

4. (Currently Amended) The computer network controller of claim 3,
2 wherein:

said context block is configured for dynamic allocation of contexts between
4 inbound remote direct memory access, inbound remote memory access and outbound
remote memory access;

6 two upper contexts are reserved for locally driven remote direct memory access;;
and

8 said context block is configured to store information including one or more of the
following events:

- 10 - expected sequence number of a next packet for sequence checking,
- input gathering size in order to optimize use of an attached bus,
- 12 - packet type defined by the network for a specific virtual channel,
- accumulated message cyclic redundancy check for data integrity,
- 14 - source addresses,
- destination addresses,
- 16 - mapping for remote direct memory access operations,
- dedicated flags to facilitate new mapping,
- 18 - word count zero detection, and
- protection tag check; and

20 wherein said events:

are received from said inbound scheduler, said micro sequencer and said
22 network protocol engine;

are synchronized by said context block; and

24 are used by said micro sequencer to invoke, restart, switch or terminate a
thread immediately.

5. (Currently Amended) The computer network controller of claim 1,

2 wherein:

4 said micro sequencer is further configured to control said network protocol
6 engine;
8 said network protocol engine is configured to perform link injection control,
based on feedback from a link layer and intervention from an operative system;; and
 said network protocol engine is further configured to schedule packets to the
network.

6. (Previously Presented) The computer network controller of claim 1,
2 wherein said inbound scheduler is configured to decode, schedule and invoke running
tasks or allocate new tasks, based on:

- 4 i) packets received from the network,
- - - - - ii) memory mapped operations received from a bus attachment module,
6 iii) descriptors inserted in first-in, first-out work queues by a user application, and
iv) tasks received from said context block.

7. (Currently Amended) In a system area network comprising a plurality of
2 host channel adapters, a plurality of target channel adapters and a switching fabric, each
said adapter comprising:

4 a network protocol engine configured to schedule packets for transmission onto
the system area network;

6 a data buffer configured to handle one or more payloads;
 a fully associative context block configured to hold a plurality of last recently
8 used contexts to provide a dynamic resource allocation scheme reflecting run time
situations; ~~and~~

10 an address translation table coupled to said network protocol engine and
configured to:

12 maintain inbound address mapping; and

store context information not currently stored in said context block; and

14 a dedicated, programmable micro sequencer tightly coupled to said context block
and configured to;

16 control said context block and handle control flow; and
process multiple types of network packets and protocols;
18 a bus attachment module; and
a network link interface;
20 wherein said micro sequencer is packet format independent and network
independent, and wherein said contexts are updated by said micro sequencer, by an
22 inbound scheduler and by said a-network protocol engine,
a method for local and remote asynchronous completion control, the method
24 comprising:
detecting a final packet of a message directed from a local node to a remote node,
26 the final packet comprising:
an accumulated cyclic redundancy check covering the message; and
28 an address of a process completion queue on the remote node;
receiving the final packet at the remote node;
30 at the remote node:
performing an integrity check on the final packet;
32 signaling "receive complete" to the remote process completion queue; and
issuing a response to the final packet to the local node; and
34 at the local node, signaling "send complete" to a local process completion queue.

8. (Currently Amended) A protocol engine for a channel adapter configured
2 to interface a system area network with a network node, the protocol engine comprising:
an inbound scheduler configured to schedule one or more of the following for
4 each of a plurality of tasks: decoding, scheduling and invoking;
a multi-context micro sequencer configured to handle control flow for multiple
6 communication channels between the network node and the system area network,
wherein said multi-context micro sequencer is packet format independent and network
8 independent;
a context block configured to store a set of least recently used contexts, wherein
10 each said context corresponds to one of the communication channels;
a data buffer configured to buffer payloads of packets for the multiple

12 communication channels; and
a network protocol engine configured to schedule transmission of packets onto the
14 system area network;
wherein a subset of said set of contexts stored in said context blocks is reserved
16 for outbound RDMA (Remote Direct Memory Access); and
wherein a remainder of said contexts in said set of contexts are dynamically
18 allocated among inbound RDMA, inbound RMA (Remote Memory Access) and
outbound RMA.

9. (Previously Presented) The protocol engine of claim 8, wherein said multi-
2 context micro sequencer is further configured to:
detect page boundary crossing and word count zero; and
4 perform an integrity check of a message, wherein the message comprises one or
more packets.

10. (Previously Presented) The protocol engine of claim 8, wherein said multi-
2 context micro sequencer is further configured to perform integrated local and remote
completion.

11. (Cancelled)

12. (Cancelled)

13. (Previously Presented) The protocol engine of claim 8, wherein each said
2 context stored in said context block comprises one or more of:
a source address;
4 a destination address;
RDMA operation mapping;
6 expected sequence number of a next packet;
an accumulated cyclic redundancy check; and
8 a set of dedicated flags for performing one or more of:

word count zero detection;
10 packet integrity checking;
sequence error checking;
12 protection tag checking; and
data buffer management.

14. (Previously Presented) The protocol engine of claim 8, wherein said data
2 buffer comprises a number of entries equivalent to the number of least recently used
contexts stored in said context block.

15. (Previously Presented) The protocol engine of claim 8, wherein said data
2 buffer comprises:
multiple read ports; and
4 multiple write ports;
wherein said multiple read ports and multiple write ports facilitate processing of
6 multiple tasks in parallel by the protocol engine.

16. (Previously Presented) The protocol engine of claim 8, further comprising:
2 one or more work queues configured to store descriptors inserted by applications
executing on the network node; and
4 an inbound scheduler configured to schedule processing of said descriptors.

17. (Previously Presented) The protocol engine of claim 16, wherein said
2 inbound scheduler is further configured to schedule:
receipt of a packet from the system area network;
4 a memory-mapped operation received from the network node; and
a task received from said context block.

18. (Previously Presented) The protocol engine of claim 8, further comprising:
2 a first connection coupling the protocol engine to an internal bus of the network
node; and

4 a second connection coupling the protocol engine to the system area network.

19. (Previously Presented) The protocol engine of claim 18, further
2 comprising:

a third connection coupling the protocol engine to an address translation table;
4 wherein the address translation table is configured to:
maintain inbound address mapping; and
6 store context information not currently stored in said context block.

20. (Previously Presented) The protocol engine of claim 18, wherein the size of
2 packets exchanged between the protocol engine and the network node differ from the size
of packets exchanged between the protocol engine and the system area network.
